



* Taqueria * OAXAQUEÑO

* DESCRIPTIVO * TECNICO

* VERSION *

1.0 30/05/24

EQUIPO DE DESARROLLO

CHRISTIAN JOSE MERIDA AGUILAR FULLSTACK DEVELOPER

NORBERTO RAYMUNDO ARREAGA SÁNCHEZ FULLSTACK DEVELOPER

DANIEL EDUARDO OROZCO OROZCO SCRUM MASTER

ESVIN ALEXIS GUZMÁN MAZARIEGOS PRODUCT OWNER



1.	Introducción	3
2.	Objetivos	4
3.	Tecnología utilizada	5
4.	Requisitos del sistema.....	6
5.	Casos de Uso.....	7
6.	Descripción de actores	7
7.	Relacional Normalizado	10
8.	Diagrama E-R.....	¡Error! Marcador no definido.
9.	Diagrama de Flujo	13
10.	Diccionario de datos	14
11.	Glosario	17
12.	Bibliografía	19

1. Introducción

El presente documento proporciona la documentación técnica del software que se implementará en la taquería El Oaxaqueño, una taquería especializada en comida mexicana. El objetivo principal del sistema es automatizar y optimizar los procesos administrativos y operativos, incluyendo la gestión de productos, ventas, clientes y otras funcionalidades clave del negocio.

Este documento detalla los fundamentos tecnológicos utilizados, la arquitectura del sistema, el diseño de la base de datos y los requisitos técnicos para su implementación. El software será desarrollado utilizando Java bajo el paradigma de Programación Orientada a Objetos (POO). La base de datos estará gestionada por SQL Server, y el despliegue se realizará en un entorno local utilizando una computadora con especificaciones definidas más adelante.

2. Objetivos

Objetivo General

Desarrollar y documentar técnicamente un sistema de gestión para "El Oaxaqueño", que permita optimizar sus procesos internos y facilitar el control de ventas, productos e historial de transacciones, asegurando su correcta implementación y mantenimiento a través de una documentación completa.

Objetivos Específicos

- Describir las tecnologías empleadas (Java, SQL Server, NetBeans) y justificar su elección.
- Diseñar y presentar el modelo lógico y físico de la base de datos, incluyendo el diagrama entidad-relación.
- Establecer los requisitos de hardware y software necesarios para el despliegue y correcto funcionamiento del sistema.
- Proporcionar una base documental para futuras expansiones y actualizaciones del sistema.
- Desarrollar una interfaz gráfica en Java Swing, con acceso directo a la lógica del sistema.

3. Tecnología utilizada

1. Lenguaje de programación

El sistema está desarrollado en **Java**, un lenguaje de programación robusto, multiplataforma y orientado a objetos. Definiendo una estructura lógica basada en clases que representan entidades como Mesa y pedidos.

2. Programación Orientada a Objetos

El sistema está organizado bajo los principios de POO:

- **Encapsulamiento:** Cada clase maneja internamente sus datos.
- **Abstracción:** Las clases representan conceptos del dominio de la taquería.
- **Herencia:** Se aplica en clases genéricas como Persona, de la cual puede derivar Cliente o Usuario.
- **Polimorfismo:** Permite adaptar comportamientos entre clases relacionadas.

3. Sistema de Gestión de Base de Datos

Se utiliza **SQL Server**, por su fiabilidad y capacidad para manejar datos empresariales. La conexión se realiza mediante JDBC. Se diseñará una base de datos que refleje las entidades principales del negocio, como productos, ventas, usuarios y clientes.

4. Interfaz gráfica

La interfaz de usuario se desarrolla con **Java Swing**, permitiendo una experiencia de escritorio amigable. La interfaz se comunica directamente con las clases del modelo de negocio (sin separación tipo MVC), permitiendo una integración rápida y eficaz.

5. Entorno de desarrollo

- **IDE:** Apache NetBeans 24
- **JDK:** Java SE 17 o superior
- **JDBC Driver:** Para conexión a SQL Server
- **Control de versiones:** Git, con repositorio local o en GitHub

6. Despliegue en entorno local

El sistema está diseñado para ejecutarse de forma **local** en un equipo de escritorio o laptop dentro del taquería, sin requerir acceso a internet.

4. Requisitos del sistema

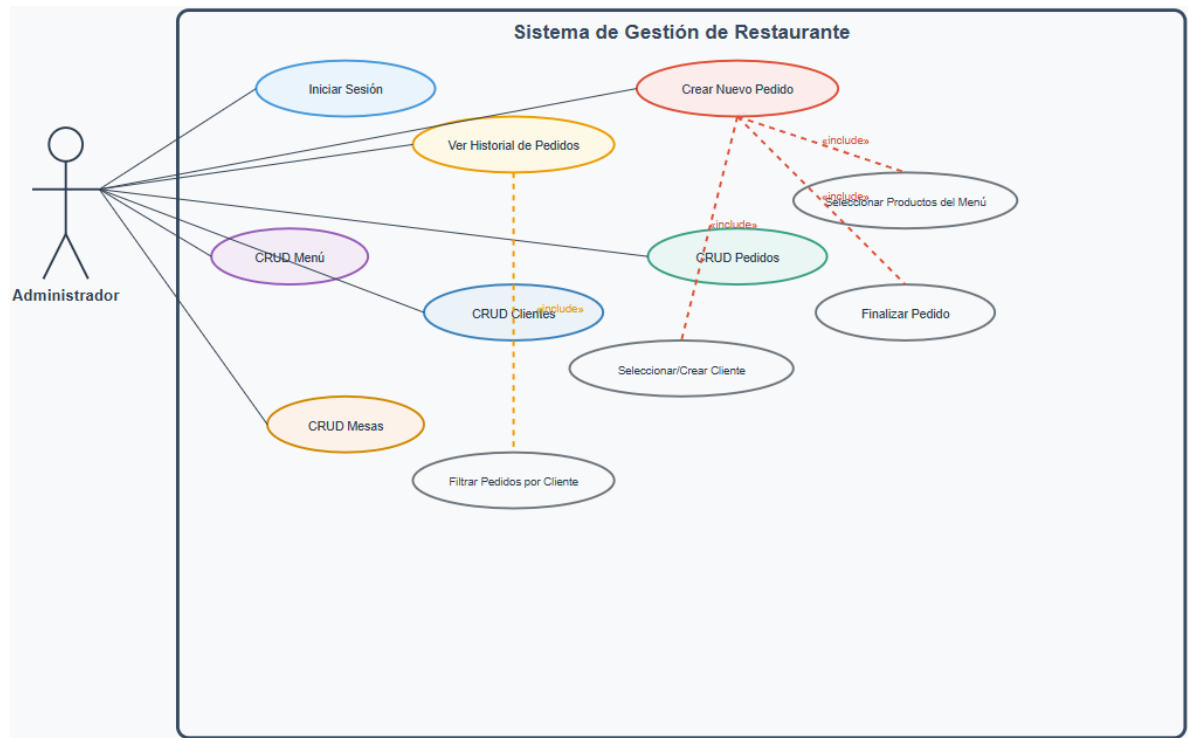
1. Hardware

- **Procesador:** Intel Core i5 (va generación) o equivalente
- **RAM:** 4 GB (recomendado: 8 GB)
- **Almacenamiento:** 250 GB (SSD recomendado)

2. Software

- **Sistema operativo:** Windows 10 o superior
- **Java JDK**
- **Apache NetBeans IDE:** versión 24
- **SQL Server:** Versión 2019 o superior
- **JDBC Driver:** Para conexión con SQL Server

5. Casos de Uso



6. Descripción de actores

Actor: Administrador

Descripción General:

El **Administrador** es el usuario clave del sistema, encargado de controlar y gestionar todas las funcionalidades operativas y administrativas de la taquería. Este actor tiene privilegios completos dentro del sistema y es responsable de asegurar que el flujo de trabajo sea eficiente, desde la gestión del menú hasta el registro de pedidos y atención al cliente.

Responsabilidades Específicas:

1. Iniciar Sesión:

- Accede al sistema mediante autenticación.
- Valida credenciales para garantizar seguridad en el acceso.

2. CRUD Menú:

- Crea nuevos productos que estarán disponibles para los pedidos.
- Modifica productos existentes (nombre, precio, categoría).
- Elimina productos que ya no están disponibles.
- Visualiza la lista completa del menú.

3. **CRUD Clientes:**

- Registra nuevos clientes en la base de datos.
- Edita la información de clientes (nombre, contacto, dirección).
- Elimina registros de clientes.
- Visualiza los datos de los clientes.
- **Nota:** Este caso de uso está vinculado al caso de uso de "Crear Nuevo Pedido", ya que se puede requerir registrar o seleccionar un cliente antes de procesar un pedido.

4. **Filtrar Pedidos por Cliente:**

- Permite al administrador buscar y listar los pedidos realizados por un cliente específico.
- Útil para reclamos, historial de consumo o seguimiento de pedidos frecuentes.

5. **CRUD Mesas:**

- Administra las mesas disponibles en la taquería.
- Asigna nombres, números o ubicaciones a las mesas.
- Realiza cambios según disponibilidad física.

6. **Ver Historial de Pedidos:**

- Visualiza todos los pedidos realizados históricamente.
- Puede incluir filtros por fecha, cliente, estado, etc.

7. **CRUD Pedidos:**

- Permite gestionar los pedidos existentes (crear, modificar, eliminar, consultar).
- Relacionado directamente con el proceso operativo diario de la taquería.

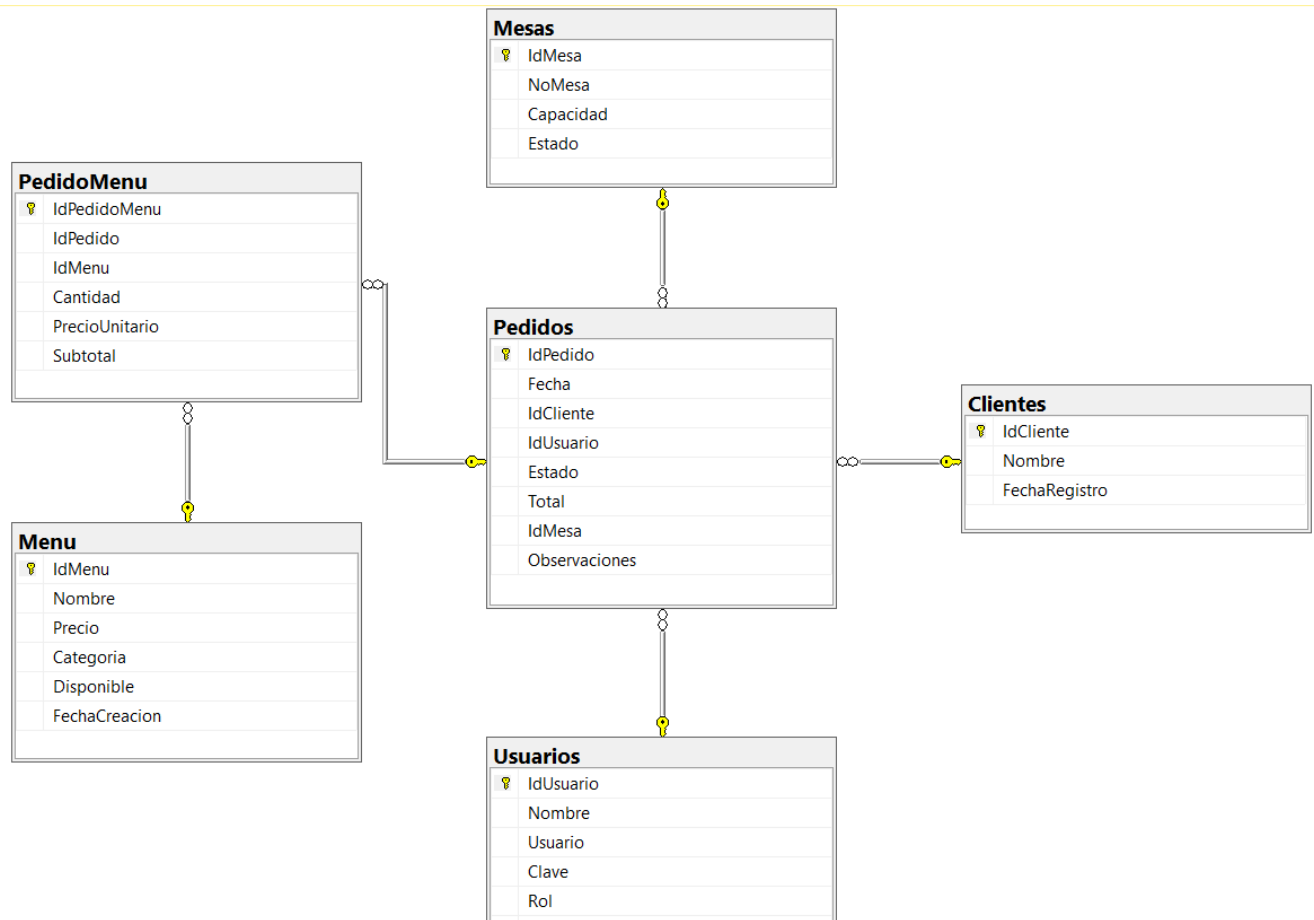
8. **Crear Nuevo Pedido:**

- Inicia un nuevo pedido desde cero.
- **Este caso de uso incluye (relación <<include>>):**
 - **Seleccionar/Crear Cliente:** El administrador elige un cliente existente o registra uno nuevo.
 - **Seleccionar Productos del Menú:** Elige los productos que el cliente desea consumir.
 - **Finalizar Pedido:** Registra y cierra formalmente el pedido, quedando listo para ser preparado y/o entregado.

Perfil del Actor:

- Rol ocupado normalmente por personal administrativo, gerente del taquería o personal autorizado.
- Tiene pleno acceso a los módulos del sistema.
- Puede consultar datos sensibles, realizar operaciones críticas y mantener actualizado el sistema.

7. Relacional Normalizado



8. Modelo físico de datos

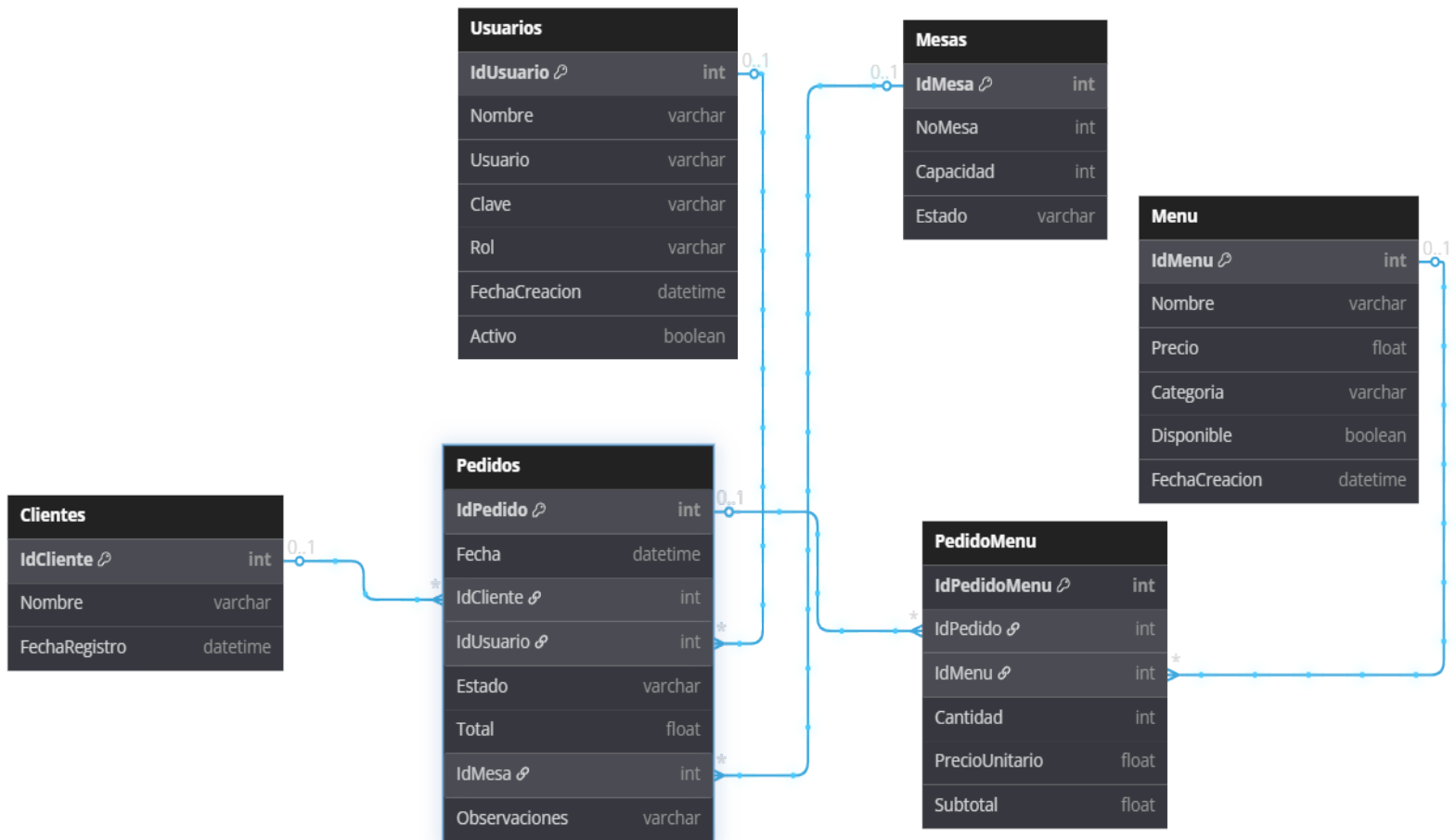
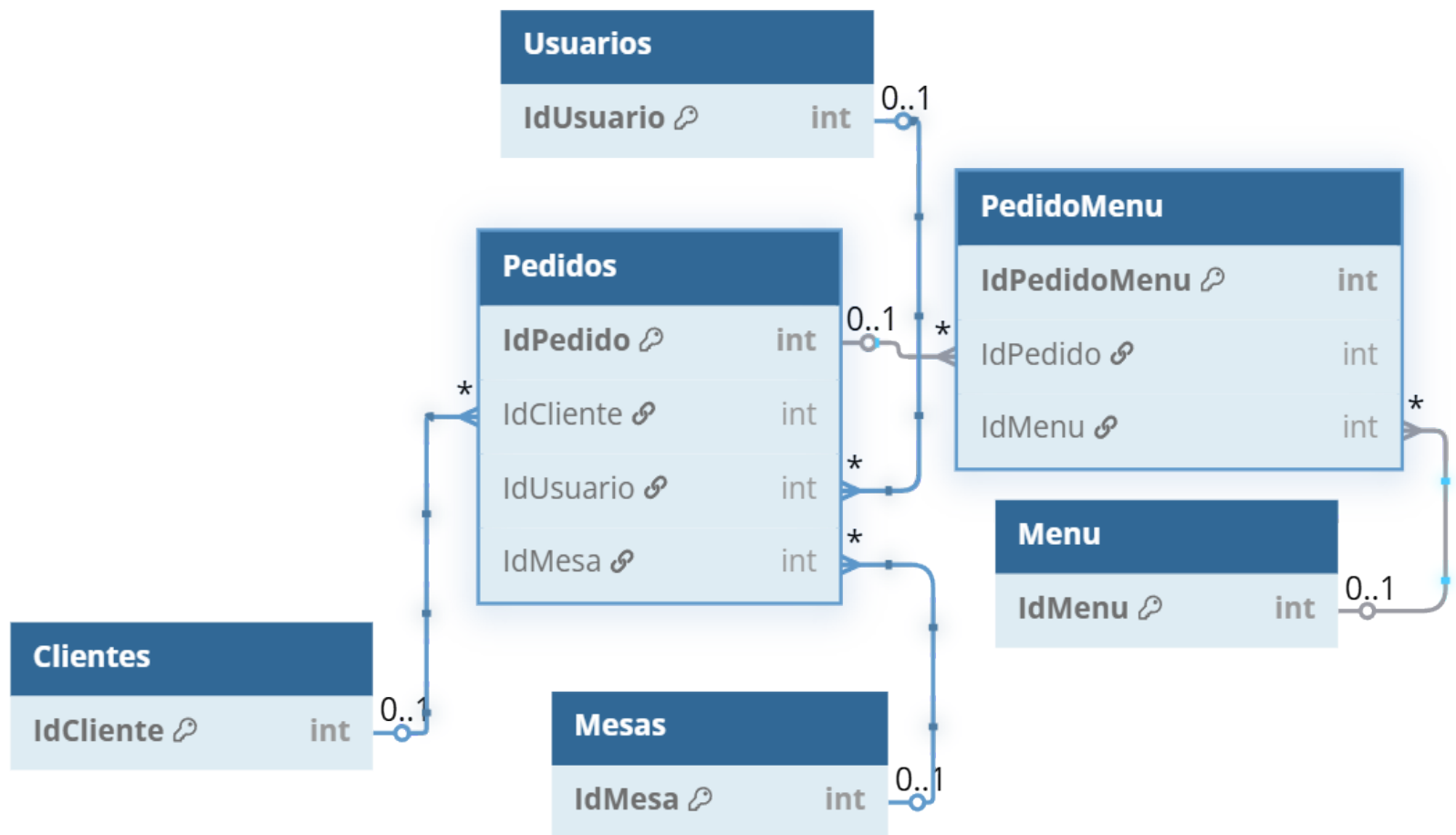
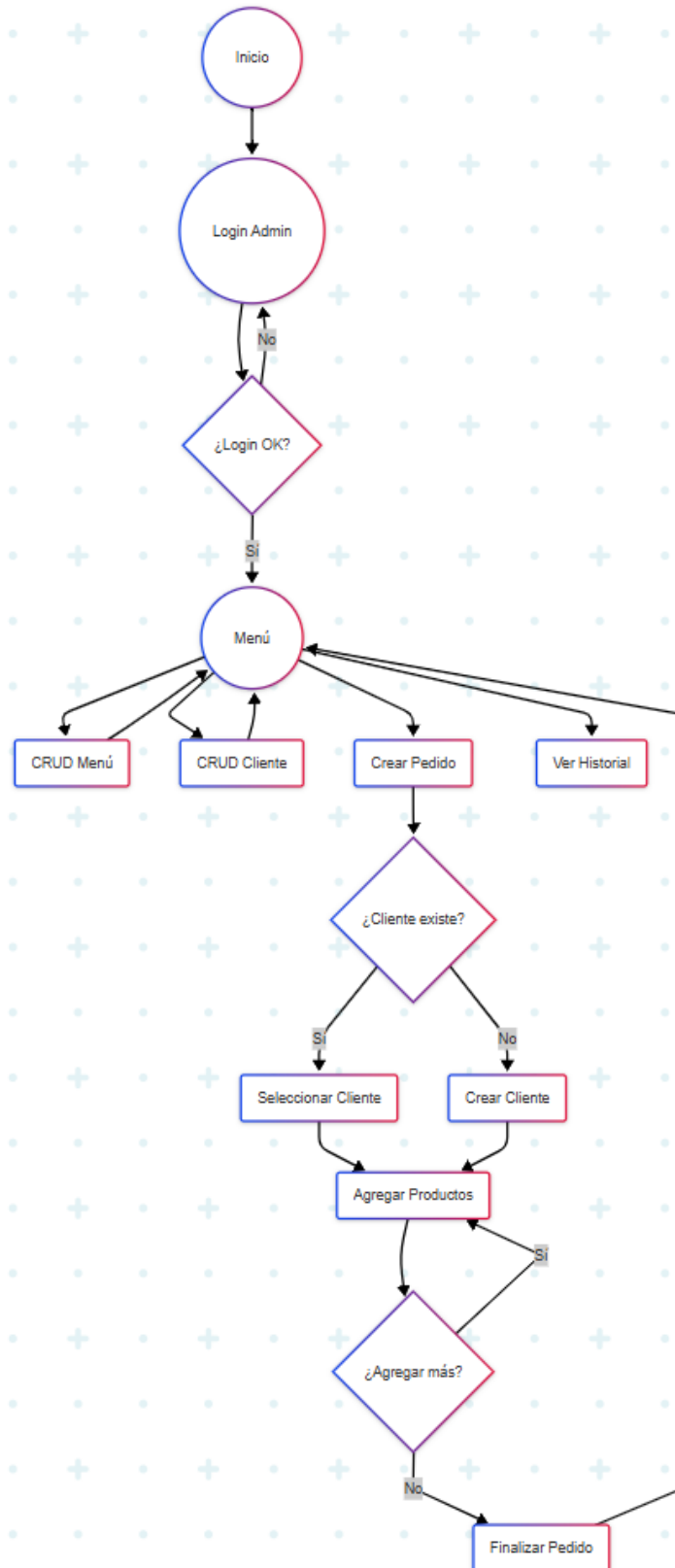


Diagrama E-R



9. Diagrama de Flujo



10.Diccionario de datos

Tabla Usuarios

Campo	Tipo de Dato	Longitud	Descripción
IdUsuario	INT (PK)	-	Identificador único del usuario
Nombre	NVARCHAR	100	Nombre completo del usuario
Usuario	NVARCHAR	50	Nombre de usuario (único para login)
Clave	NVARCHAR	100	Contraseña encriptada del usuario
Rol	NVARCHAR (CHECK)	50	Rol del usuario (Administrador, Mesero, etc.)
FechaCreacion	DATETIME	-	Fecha de creación del usuario
Activo	BIT	-	Estado activo/inactivo del usuario

Tabla Clientes

Campo	Tipo de Dato	Longitud	Descripción
IdCliente	INT (PK)	-	Identificador único del cliente
Nombre	NVARCHAR	100	Nombre del cliente
Telefono	NVARCHAR	20	Teléfono del cliente (opcional)
FechaRegistro	DATETIME	-	Fecha de registro del cliente

Tabla Mesas

Campo	Tipo de Dato	Longitud	Descripción
IdMesa	INT (PK)	-	Identificador único de la mesa
NoMesa	INT	-	Número de la mesa (único)
Estado	NVARCHAR (CHECK)	20	Estado actual (Disponible, Ocupada)

Tabla Menu

Campo	Tipo de Dato	Longitud	Descripción
IdMenu	INT (PK)	-	Identificador único del platillo
Nombre	NVARCHAR	100	Nombre del platillo o bebida
Precio	DECIMAL(10,2)	-	Precio unitario del producto
Categoria	NVARCHAR (CHECK)	50	Categoría del menú (Tacos, Especialidades, etc.)
Disponible	BIT	-	Si el producto está disponible
FechaCreacion	DATETIME	-	Fecha en que se agregó al sistema

Tabla Pedidos

Campo	Tipo de Dato	Longitud	Descripción
IdPedido	INT (PK)	-	Identificador único del pedido
Fecha	DATETIME	-	Fecha y hora del pedido
IdCliente	INT (FK)	-	Cliente que realizó el pedido
IdUsuario	INT (FK)	-	Usuario que registró el pedido
Estado	NVARCHAR (CHECK)	50	Estado del pedido (Pendiente, Entregado, etc.)
Total	DECIMAL(10,2)	-	Total a pagar del pedido
IdMesa	INT (FK)	-	Mesa asignada al pedido
Observaciones	NVARCHAR	500	Notas o comentarios del pedido

Tabla PedidoMenu

Campo	Tipo de Dato	Longitud	Descripción
IdPedidoMenu	INT (PK)	-	Identificador del producto dentro del pedido
IdPedido	INT (FK)	-	Pedido al que pertenece
IdMenu	INT (FK)	-	Producto del menú elegido
Cantidad	INT	-	Número de unidades solicitadas
PrecioUnitario	DECIMAL(10,2)	-	Precio unitario del producto al momento del pedido
Subtotal	CALCULADO (PERSISTED)	-	Cantidad * PrecioUnitario

11.Glosario

- **Java:** Lenguaje de programación orientado a objetos utilizado para el desarrollo del sistema.
- **JDK:** Conjunto de herramientas necesarias para compilar y ejecutar aplicaciones Java.
- **POO:** Paradigma de programación que organiza el código en clases y objetos para representar conceptos del mundo real.
- **Clase:** Estructura que define atributos y comportamientos comunes a un tipo de objeto en el sistema.
- **Objeto:** Instancia concreta de una clase que se utiliza en tiempo de ejecución.
- **Atributo:** Variable definida dentro de una clase que representa una propiedad del objeto.
- **Método:** Función definida dentro de una clase que representa un comportamiento del objeto.
- **Encapsulamiento:** Técnica para ocultar los detalles internos de una clase, permitiendo acceso controlado mediante métodos públicos.
- **Herencia:** Permite crear nuevas clases basadas en clases existentes, reutilizando código.
- **Polimorfismo:** Capacidad de que el método se comporte de forma diferente según el objeto que le invoque.
- **Abstracción:** Principio que permite representar entidades del sistema con solo los datos y comportamientos esenciales.
- **NetBeans:** Entorno de desarrollo integrado (IDE) utilizado para programar en Java.
- **Swing:** Biblioteca gráfica de Java usada para construir interfaces gráficas de escritorio (GUI).
- **SQL Server:** Sistema de gestión de base de datos relacional utilizado para almacenar la información del sistema.
- **JDBC (Java Database Connectivity):** API de Java que permite la conexión y ejecución de operaciones en bases de datos relacionales como SQL Server.
- **Base de Datos Relacional:** Conjunto estructurado de datos organizados en tablas con relaciones entre sí, basado en claves primarias y foráneas.
- **Tabla:** Estructura de almacenamiento dentro de una base de datos, compuesta por filas (registros) y columnas (campos).
- **Clave Primaria:** Campo único que identifica de forma exclusiva cada registro dentro de una tabla.
- **Clave Foránea:** Campo que crea una relación entre dos tablas referenciando la clave primaria de otra tabla.
- **Consulta SQL:** Instrucción escrita en lenguaje SQL para obtener, insertar, actualizar o eliminar datos de la base de datos.
- **Driver JDBC:** Componente que permite la comunicación entre una aplicación Java y una base de datos como SQL Server.

- **Interfaz Gráfica (GUI):** Entorno visual interactivo que permite al usuario comunicarse con el sistema sin usar línea de comandos..
- **Persistencia de Datos:** Proceso de almacenamiento permanente de información en una base de datos, evitando su pérdida al cerrar el programa.
- **Sistema Monousuario:** Aplicación diseñada para ser utilizada por un solo usuario al mismo tiempo en un solo equipo.

12.Bibliografia

Oracle. (2024). *The Java™ Tutorials*. <https://docs.oracle.com/javase/tutorial/>

Pressman, R. S. (2014). *Ingeniería del software: Un enfoque práctico* (7.^a ed.). McGraw-Hill.

Date, C. J. (2004). *Introducción a los sistemas de bases de datos* (8.^a ed.). Pearson Educación.

Microsoft. (2024). *SQL Server Documentation*. <https://learn.microsoft.com/en-us/sql/sql-server/>

Oracle. (2024). *JDBC Basics*. <https://docs.oracle.com/javase/tutorial/jdbc/basics/>

W3Schools. (2025). *Tutorial de Java*. <https://www.w3schools.com/java/>